Comprehensive User Guide: Trading Bot Installation and Deployment

June 13, 2025

Contents

1 Overview

This user guide provides step-by-step instructions for setting up, configuring, and deploying a trading bot that:

- Automates trading using the Interactive Brokers (IB) API
- Generates trade summaries via Google Gemini API

It is designed for developers using Visual Studio Code (VS Code) as the primary IDE, with alternative instructions for other environments like PyCharm, terminal, and Docker.

2 Prerequisites

- **Directory Contents:** Ensure the directory includes:

```
main.py, order.py, llm_reporter.py, signals.py, portfolio.py,
fetch_data.py, indicators.py, checks.py, reporting.py,
llmcheck.py, logger.py, config.py, requirements.txt
```

- **Operating System:** Windows, macOS, or Linux
- **Python:** Version 3.8 or higher
- **Interactive Brokers:** Install TWS or IB Gateway
- **Google Gemini API Key:** Required for generating summaries
- **Internet Connection:** Required for downloading packages and calling APIs

IB Gateway Demo Account:

- macOS: `https://download2.interactivebrokers.com/installers/ibgateway/latest-standalone/ibgateway-latest-standalone-macos-arm.dmg`
- Windows (64-bit): `https://download2.interactivebrokers.com/installers/ibgateway/latest-standalone/ibgateway-latest-standalone-windows-x64.exe`

3 Core Installation (VS Code)

3.1 Step 1: Set Up VS Code

1. Download and install VS Code from `https://code.visualstudio.com`
2. Install the Python extension:
   - Open Extensions: `Ctrl+Shift+X` (Windows/Linux), `Cmd+Shift+X` (macOS)
   - Search and install: `Python` by Microsoft
3. Open project folder: `File > Open Folder`, then select your bot directory

3.2 Step 2: Set Up Python Environment

1. Verify Python version:

```
python --version
```

2. Navigate to project directory:

```
cd /path/to/trading-bot
```

3. Create and activate virtual environment:

   - Windows:

```
python -m venv venv
.\venv\Scripts\activate
```

   - macOS/Linux:

```
python -m venv venv
source venv/bin/activate
```

4. Select interpreter in VS Code:

   - `Ctrl+Shift+P` (or `Cmd+Shift+P`) → `Python: Select Interpreter` → choose `./venv/bin/python`

3.3 Step 3: Install Dependencies

1. Ensure `requirements.txt` includes:

```
dotenv
numpy
pandas
ibapi>=9.81.1
ib_insync
matplotlib
google-generativeai>=0.8.1
python-dotenv
```

2. Install dependencies:

```
pip install -r requirements.txt
```

3. Verify installation:

```
pip list
```

   Ensure the following are installed: `google-generativeai, ibapi, ib_insync, python-dotenv`

3.4 Step 4: Configure Interactive Brokers API

1. Upgrade IB packages:

```
pip install --upgrade ibapi ib_insync
```

2. Launch TWS or IB Gateway:

   - Go to `File > Global Configuration > API > Settings`

- Enable API access
- Set port to `4002` (paper) or `7496` (live)
- Allow connections from `127.0.0.1`

3. In `config.py`, confirm:

```
IB_HOST = "127.0.0.1"
IB_PORT = 4002
IB_CLIENT_ID = 1
```

3.5 Step 5: Configure Gemini API

1. Install Gemini API:

```
pip install "google-generativeai>=0.8.1"
```

2. Generate API key:

   - Visit `https://aistudio.google.com/app/apikey`
   - Create/select project and enable Gemini API
   - Copy your API key

3. Create `.env` file:

```
GEMINI_API_KEY=your-gemini-api-key
```

4. Load the environment variable in `main.py`:

```
from dotenv import load_dotenv
import os
load_dotenv()
api_key = os.getenv("GEMINI_API_KEY")
```

3.6 Step 6: Verify and Prepare Files

- Ensure all project files are present
- Confirm `config.py` values:

```
IB_PORT = 4002
SYMBOLS = ['NET']
ORDER_TYPE = "AZN"
STOP_LOSS_PCT = 0.05
TAKE_PROFIT_PCT = 0.10
```

- Logger: Ensure `logger.py` writes to logs and console
- Create directories if missing:

```
mkdir -p charts reports logs
```

3.7 Step 7: Test Setup

1. Run:

```
python llmcheck.py
```

2. Fix common issues:
   - Validate `.env` and key
   - Ensure TWS/Gateway is active

3.8 Step 8: Deploy and Run

1. Start TWS/Gateway and log in
2. In VS Code:
   - Open `main.py`, right-click → Run Python File in Terminal
3. Monitor:
   - Logs in terminal and `logs/`
   - Charts in `charts/`, reports in `reports/`

4 Operating the Bot

- **Monitoring:** Logs, charts, and trade reports
- **Trading Logic:**
  - Buy: Price > SMA, RSI < oversold
  - Sell: Price < SMA or RSI > threshold
  - Short: Price < RSI or > RSI, Cover accordingly
- **Checks:** Validate buying power, shortability
- **Portfolio:** Tracks positions, logs P&L

5 Best Practices

- Use paper trading before live deployment
- Test manually using `buystock.py` if needed
- Monitor logs continuously
- Adjust parameters:
  - `DEFAULT_POSITION_SIZE`, `LIMIT_PRICE_OFFSET`
- Keep dependencies and TWS/Gateway updated